



Plataforma Java

Transcripción

[00:00] Hola a todos, sean bienvenidos. Vamos a comenzar hablando un poco sobre la motivación de por qué aprender Java. Muchos de repente ya tienen un background en lenguaje de programación, puede ser Python, Basic, Scala, no lo sé.

[00:21] Quizás algunas de las características de Java que en este caso pues es un lenguaje orientado a objetos, es un lenguaje fácil de entender, es multiplataforma, quizás algunos otros lenguajes de programación ya tienen este feature o estos features, pero la ventaja de Java no es el lenguaje en sí. La principal ventaja de Java es la plataforma.

[00:47] Es la plataforma lo que nos va a servir y va a hacer que Java sea un lenguaje diferenciado y que nos va a abrir un conjunto de oportunidades prácticamente infinitas. Recordemos que incluso hay lenguajes que se basan en Java en la actualidad, pero vamos por partes.

[01:06] La plataforma Java básicamente sería la máquina virtual de Java que es lo principal, los distintos frameworks que tenemos, las distintas librerías de la comunidad open source totalmente gratuitas que tenemos para el uso y la facilidad para implementar todo ello.

[01:27] Entre las principales características que tenemos de la plataforma Java es portable, porque puede ser cargada en cualquier dispositivo, es fácil de implementar, es segura y es omnipresente. Algunos ejemplos que tenemos para describir mejor estas características son: encontramos Java en servidor

la máquina virtual de Java está presente en servidores de bancos, empresas grandes, enormes, justo por las características que tiene.

[01:57] Está presente en dispositivos móviles, teléfonos, tablets, hay una infinidad de dispositivos que corren Java, disponible también en computadores como el mío actualmente, como el de ustedes también. Hoy en día ya básicamente es imposible tener un computador sin Java porque de hecho muchas de las aplicaciones actuales necesitan la máquina virtual de Java para ejecutarse.

[02:22] Encontramos en dispositivos de uso común, como impresoras, computadoras, televisores, reproductores de VHS, DVD, y aparte de eso más de 100 millones de descargas anuales de la máquina virtual de Java. básicamente la fuerza de Java radica en esto.

[02:45] Pero vamos por partes. Vamos a un poco hablar de la máquina virtual de Java en sí y cómo surgió la idea de crear esta máquina virtual de Java. Vamos allá atrás a los años 1960, 1970, cuando teníamos distintos tipos de dispositivos. En este caso, solo por dar un ejemplo tenemos dispositivos de juegos, televisores y en ese tiempo, VHS. Para el que no conoce el VHS, es el antecesor del DVD, del Blu-ray.

[03:17] Entonces básicamente el proceso de desenvolvimiento del software para estos dispositivos era escribir código, generar un ejecutable. En este caso por ejecutable entendemos que es aquel código ya traducido a lenguaje entendible por la máquina que va a ser ejecutado por la máquina, por el procesador, por el fierro, y finalmente ejecutarlo, correrlo en el dispositivo final.

[03:47] Si tú querías hacer un código para una consola de videojuegos, esa consola tenía su propio lenguaje, su propia sintaxis, y no era compatible con el lenguaje y la sintaxis de la TV y de igual forma con algunos otros dispositivos. Esto se aplicaba para cada uno de los dispositivos.

[04:06] Recordemos que en esos tiempos la mayoría de empresas generaban ingresos gracias a la venta de hardware, generación de hardware, y había una empresa, Sun Microsystems, que se dedicaba a esto, venta de servidores, todo esto, y el software venía gratis.

[04:25] En Sun Microsystems trabajaba un señor muy conocido en el mundo del software llamado James Gosling. James Gosling es el padre de Java, es considerado el padre de Java, pero no fue él solo quien inventó Java, fue él con un conjunto de ingenieros de Sun Microsystems que vieron este problema de aquí y dijeron: "¿No sería mejor solamente quedarnos con los comunes denominadores y generar un dispositivo universal que se encargue de traducir aquel código?"

[05:01] Ya sea para un juego, una TV o algún otro dispositivo, y solamente escribir un solo código y generar un solo ejecutable, algo universal. ¿Qué idea se les ocurrió? Crear una máquina, un dispositivo físico. Esa fue la primera versión de la máquina virtual de Java, un dispositivo físico, que la idea era tener algo que parezca un computador, que se encargue de traducir esas instrucciones y ejecutarlo en cualquier dispositivo, hacerlo omnipresente.

[05:34] Obviamente esto generaba un costo extra para la marca que está desarrollando aquel dispositivo y obviamente fracasó. Hoy en día nadie conoce un dispositivo con esa función, solamente de interpretar código y ejecutar. Entonces, después de un tiempo que la idea fracasó, tuvieron otra gran idea que fue, ya viendo cómo era el proceso de desarrollo de software de Microsoft.

[06:02] En este caso tenemos que recordar que si bien en esa época las empresas ganaban dinero vendiendo hardware, poco a poco IBM y Microsoft fueron saliendo de esa curva y comenzaron a generar dinero vendiendo software, y fue ahí cuando dijeron: "Nosotros también necesitamos ganar dinero con software".

[06:22] Entonces vemos aquí que el lenguaje de código y el lenguaje intermedio eran los comunes denominadores de igual forma para Windows, era el mismo proceso, solamente que Windows como tenía su propio sistema operativo y en ese tiempo también acaparó la gran cantidad del mercado, era muy fácil para ellos producir software de esa manera.

[06:50] Pero al mismo tiempo también salió la competencia, salió Apple y salió Linux. En este caso dije el orden inverso porque primero fue Linux y después fue Apple, porque fue basado en Linux. Y el proceso era el mismo que para los dispositivos físicos, y ya estábamos aquí, en este momento estábamos en la guerra pues de los sistemas operativos, que sigue hasta el momento.

[07:16] Entonces ahí dijeron: "No necesitamos un dispositivo físico o una tarjeta física para cada computador con Windows, Mac y Linux. En realidad necesitamos algo que entienda esas instrucciones y sepa en qué sistema operativo está y que traduzca esas instrucciones de un solo lenguaje para la plataforma en la cual se esté corriendo". Entonces no podía ser física, así que decidieron hacer una máquina virtual.

[07:47] Y esa sería la primera Java Virtual Machine como la conocemos hoy en día. La Java Virtual Machine tenía la ventaja de que podíamos escribir código en un solo lenguaje, que en este caso era el lenguaje Java, generar un ejecutable de aquel lenguaje, que en este caso era el Bytecode.

[08:09] Bytecode es digamos el resultado del lenguaje Java, pero traducido por la máquina, por el proceso de compilación. Y él generaba un archivo punto class. Ese archivo era un conjunto de instrucciones entendible por la máquina virtual de Java, es un lenguaje muy parecido a Ceylon. Y la máquina virtual de Java ya era la encargada de entender el Bytecode y enviar las instrucciones necesarias al sistema operativo directamente.

[08:45] Entonces con esto nosotros ya teníamos cubierto el desarrollo de software para cualquier plataforma con la máquina virtual de Java instalada,

que la gran ventaja fue que no era un dispositivo físico, era un software gratuito que cualquier sistema operativo podía tener.

[09:00] Entonces, yendo un poco más a estos dos primeros puntos de lenguaje Java y Bytecode, tenemos aquí un ejemplo de una clase en lenguaje Java. Por clase entendemos que es un archivo, es un archivo en este caso refiriéndose a una persona, donde tiene los atributos nombre, apellido y edad. Y tiene un pequeño método ahí llamado datos, por defecto, en el cual asignamos nombre, apellido y edad.

[09:34] Este código es humanamente entendible, humanamente legible, no tiene alguna sintaxis extraña, quizás algunas cosas como class, void, package, mas no es nada difícil de entender. Y el Bytecode generado de esta clase sería este de aquí. Vemos que es un lenguaje mucho más robótico, es un lenguaje más entendible más para la máquina, es el lenguaje que va a ser leído e interpretado por la Java Virtual Machine.

[10:11] Ella va a entender mucho mejor este lenguaje que tenemos aquí al lado derecho y va a ejecutar las instrucciones en el ordenador, directamente basado en este archivo. Este archivo es el resultado del proceso de compilación de la clase persona que tenemos a la izquierda.

[10:31] Entonces, ¿será que la máquina virtual de Java solo compila código? No. La máquina virtual de Java se encarga de la administración de memoria, tenemos el famoso garbage collector, que lo que hace básicamente es que los archivos que dejamos de usar simplemente los borra, los apaga de la memoria y seguimos trabajando normal.

[10:53] Es multiplataforma, es segura, utiliza constantemente el uso del procesador y la gran cantidad de librerías utilitarias que tiene. Entonces, algunos lenguajes basados en Java, como les dije al inicio de la presentación, Clojure, Ruby, Scala Python, Groovy, todos ellos son soportados en la máquina virtual de Java, que genera Bytecode.

[11:17] Y la máquina virtual de Java al mismo tiempo es soportada por la mayoría de los sistemas operativos que tenemos actualmente. Entonces, tenemos ya un poco del fundamento de dónde sale la máquina virtual de Java, cuál es su uso que le damos día a día y la gran gama de oportunidades que nos da actualmente.

[11:44] Esto ha sido todo por ahora, ya nos vemos en la siguiente clase. Y recuerden: Java nos abre un conjunto infinito de oportunidades infinito gracias a su plataforma. Chau, chau.